Naval Research Laboratory
Washington, DC 20375-5320

NRL/6910/MR—2024/2

SDAB_GUI: Graphical User Interface for Novel Single-Domain Antibody Generation Using Generative Pre-Trained Transformers and Recurrent Neural Networks

JEROME ANTHONY E. ALVAREZ

SCOTT N. DEAN

Laboratory for Bio/Nano Science and Technology Branch
Center for Bio/Molecular Science and Engineering Division

March 18, 2024

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
18-03-2024	NRL Memorandum Report	START DATE 04-01-2022	END DATE 02-13-2024

4. TITLE AND SUBTITLE

SDAB_GUI: Graphical User Interface for Novel Single-Domain Antibody Generation Using Generative Pre-Trained Transformers and Recurrent Neural Networks

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
		1P79

6. AUTHOR(S)

Jerome Anthony E. Alvarez, and Scott N. Dean

7. PERFORMING ORGANIZATION / AFFILIATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Ave SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/6910/MR—2024/2

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR / MONITOR'S ACRONYM(S) NUMBER	11. SPONSOR / MONITOR'S REPORT NUMBER(S)
Defense Threat Reduction Agency 8725 John J Kingman Rd Ste 6201 Fort Belvoir, VA 22060	DTRA	

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTAL NOTES

14. ABSTRACT

This report is an update to the first version of the submitted work, Graphical User Interface for Novel Protein Generation using ProtGPT: Version 1, which detailed the Year 4 efforts of the Naval Research Laboratory (NRL) toward the algorithm development for optimization of biologic medical countermeasures.

The directed modification of antibody proteins and the foundational implications of reengineering and synthesizing of novel, small molecules with desired properties are further developed in modern biotechnology to combat antibiotic resistance. Specifically, the expensive and production delays of novel single-domain antibodies (sdAbs) pose the challenge in antibiotic discovery. This, however, can be mitigated with fast, efficient, and automated sdAb sequence generation ready to be synthesized for potential therapeutic use. Thus, we developed SDAB_GUI – a ready-to-use deployable software with a graphical user interface (GUI) using an open-source, interactive application which stemmed from a previously reported prototype. SDAB_GUI incorporates generative pre-trained transformer (GPT) and recurrent neural network (RNN) models for sdAb generation and enables users to follow a set of cascaded steps for the input, exploration, and generation of new sequences based on an existing sdAb dataset. Summary statistics of the input and output sdAb sequence dataset, exploratory analyses, and the proteins' physicochemical characteristics can also be interactively employed. Validation measures are included along the sequence generation such as redundancies or antibody numbering annotations to produce proper sdAb characteristics. This automated process allows the generation of novel sdAb sequences that can subsequently be evaluated for improvements to a specifically desired biophysical property.

This document also serves as the reference material for the SDAB GUI software.

15. SUBJECT TERMS

antibodies; proteins; graphical user interface; generative deep learning; generative pre-trained transformer, recurrent neural networks

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U	SAR	23

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
Jerome Alvarez	(202) 767-0394

This page intentionally left blank.

Table of Contents

This page intentionally left blank.

Executive Summary

This report is an update to the first version of the submitted work, Graphical User Interface for Novel Protein Generation using ProtGPT: Version 1, which detailed the Year 4 efforts of the Naval Research Laboratory (NRL) toward the algorithm development for optimization of biologic medical countermeasures.

The directed modification of antibody proteins and the foundational implications of reengineering and synthesizing of novel, small molecules with desired properties are further developed in modern biotechnology to combat antibiotic resistance. Specifically, the expensive and production delays of novel single-domain antibodies (sdAbs) pose the challenge in antibiotic discovery. This, however, can be mitigated with fast, efficient, and automated sdAb sequence generation ready to be synthesized for potential therapeutic use. Thus, we developed SDAB_GUI – a ready-to-use deployable software with a graphical user interface (GUI) using an open-source, interactive application which stemmed from a previously reported prototype. SDAB_GUI incorporates generative pre-trained transformer (GPT) and recurrent neural network (RNN) models for sdAb generation and enables users to follow a set of cascaded steps for the input, exploration, and generation of new sequences based on an existing sdAb dataset. Summary statistics of the input and output sdAb sequence dataset, exploratory analyses, and the proteins' physicochemical characteristics can also be interactively employed. Validation measures are included along the sequence generation such as redundancies or antibody numbering annotations to produce proper sdAb characteristics. This automated process allows the generation of novel sdAb sequences that can subsequently be evaluated for improvements to a specifically desired biophysical property.

This document also serves as the reference material for the SDAB GUI software.

This page intentionally left blank.

1. Introduction

The search for therapeutic and diagnostic antibodies is a growing interest in the development of biomedical research. Specifically, these proteins continue to be relevant in the production of antibiotic agents that aim to target a vast range of diseases and the overall improvement in the human quality of life. However, harmful biomolecules such as bacteria and fungi have the ability to develop an adaptation to drugs due to misuse, and even presumed to be abused through unregulated prescription, extensive agricultural use [1]. When novel antibody-derived agents are eventually used as aid or a prospective countermeasure, the emergence of its resistant counterpart is unpreventable, albeit bacterial evolution being uncertain and antibiotic resistance inevitable [2].

One of the first studies in domain antibodies was conducted in 1989 when Ward and colleagues [3] reported the isolation of stable mouse antibody VH domains that could bind antigens with relatively high affinity. In emergence from these studies, variable heavy-chain antibodies (VHHs) were found in camels in 1993 by Hamers-Casterman and co-workers [4] which represent the smallest naturally derived antigen-binding fragments approximately 15 kilodaltons (kDa). More importantly, new insights were gained by determining complexities of full-length antibodies, such as from sharks, camelids and humans to single functional fragments known as single-domain antibodies (sdAbs). These single domains can be remodeled, engineered and reassembled to generate new antibody-like molecules with desired properties and specificities, consisting of multiple units of the same or of different function [5]. Due to their smaller size than conventional antibodies, sdAbs have the advantage to access recessed epitopes [5], blood clearance and tissue accessibility [6], and better penetration of solid tumors [7]. Hence, these sdAbs' antigen-binding abilities and affinities which are relatively straightforward to engineer and more economical to produce are great candidates for antibody-based therapeutics.

As modern biotechnology makes a breakthrough in protein sequence generation through generative deep learning methods or artificial intelligence, the prominent interest in antibody production has its merits. Automated systems can process small or large sequence data through artificial neural networks that rely on many layers of nonlinear processing units for learning data representations. This architecture has been proven useful in protein synthesis molecular prediction algorithms – antimicrobial design [8], antibiotic discovery [9], and improved thermostability. While such approaches continue to develop, these systems will continue to contribute to the design and evolution of antibodies.

To aid in these efforts, appropriately engineered biomolecules should exhibit pharmacokinetic properties that make them desirable to both production and application. The design and generation of novel antibody sequences, or by production of environmentally existing variants, should be capable in combating the preceding antibiotic resistance among other attributable biophysical properties. The fast and efficient generation of single-domain antibodies (sdAbs) should be effortlessly accessible to ensure availability of novel proteins for medical countermeasures. Therefore, we developed a graphical user interface (GUI) called SDAB_GUI for antibody-based drug design using modified generative pre-trained transformer (GPT) and recurrent neural network (RNN) models. SDAB_GUI is a software application developed from its first prototype [10] which enables users, both developers and front-end clients, to follow a set of cascaded steps for the input, exploration, and generation of new sequences based on an existing antibody dataset.

Manuscript approved March 15, 2024.

2. Materials and Methods

2.1 Software

Upon receipt of the SDAB_GUI software, the user will be presented with a set of files and folders (**Figure 1**). The instructions to initialize and run the software are discussed in **Section 3: Step-by-Step Instructions**.

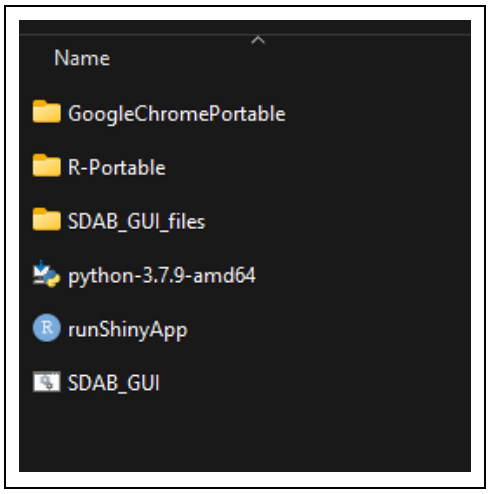

Figure 1. SDAB_GUI software folder contents. **SDAB_GUI_files is the current working directory.**

2.1.1 Portable Applications

SDAB_GUI uses portable software applications that stem from fully open-source platforms. The interface launches portable Google Chrome as the front-end and uses portable R software and a native Python installation as back-end processes.

Chrome	https://portableapps.com/apps/internet/google_chrome_portable
R Portable	https://sourceforge.net/projects/rportable/

2.1.2 Python 3.7.9

SDAB_GUI application requires an installation of Python version 3.7.9. The executable file is included in the SDAB_GUI software package.

2.1.3 Shiny by R Package

Shiny is an R Software package that is designed to build interactive applications which executes an R code as its backend (canonical form: https://CRAN.R-project.org/package=shiny). The GUI described here is built on *Shiny* application executable through the R software library and uses a portable Google Chrome as launch interface. This open-source R package provides a framework for the GUI assembly which also allows the developer to host a standalone application that consists of reactive inputs and outputs. An input can also be another third-party object-oriented programming script such as C++ or Python wherein the *Shiny* app can execute provided that the compilers for third-party languages are installed in the system.

2.1.4 System Requirements

It is important to note that SDAB_GUI only works on Windows 10/11 with a stable internet connection. R-specific libraries are pre-installed. Other dependencies such as Python module versions will be installed upon opening of software. These modules are highlighted in **Table 1**.

Table 1. Python modules and versions upon installation.

Module	Version	Reference
accelerate	0.12.0 or newer	https://huggingface.co/docs/accelerate/index
torch	1.3 or newer	https://pytorch.org/
datasets	1.8.0 or newer	https://huggingface.co/docs/datasets/v1.15.1/installation.html
sentencepiece	0.1.92 or newer	https://github.com/google/sentencepiece
protobuf	3.20.0	https://protobuf.dev/
evaluate	–	https://huggingface.co/docs/evaluate/index
scikit-learn	–	https://scikit-learn.org/stable/
transformers	–	https://huggingface.co/docs/transformers/index
tensorflow	2.10.0	https://www.tensorflow.org/
textgenrnn	–	https://github.com/minimaxir/textgenrnn
modlamp	–	https://modlamp.org/
seaborn	–	https://seaborn.pydata.org/
pytest	–	https://docs.pytest.org/en/7.4.x/

2.2 Datasets

In-house, curated single-domain antibody sequences (sdAbs) from multiple studies were used as input data. This sdAb database consists of unique 565 antibodies with corresponding sequence IDs, organism source, experimentally measured melting temperatures, target protein, framework (FR) and complementarity-determining regions (CDR), and digital object identifier source information. All sequences were restricted to the 20 natural amino acids and sequences with non-conventional residues were excluded.

2.3 Deep-Learning Models

2.3.1 Protein Generative Pre-Trained Transformer (ProtGPT) Model

A modified GPT2 transformer model from a recently published paper by Ferruz et al [11] through the Hugging Face Artificial Intelligence Group (https://huggingface.co/nferruz/ProtGPT2). Because ProtGPT2 is originally a natural language processing (NLP) transformer [12] model, it has been trained using 50 million non-annotated sequences spanning the entire protein space resulting into a "*learned*" protein language. Concomitantly, it can generate sequences that are distantly related to naturally occurring proteins and whose structures resemble the known structural space and can also sample any region, such as all-β structures and membrane proteins.

2.3.2 Recurrent Neural Networks (RNN) Model

The concept of Recurrent Neural Networks (RNN) was introduced in the early work of David Rummelhart in 1986 to improve the memory of a network by back-propagation, especially when working with sequential data [13]. The RNN model used in this software utilizes the *textgenrnn* Python module included in Table 1. Specifically, the RNN for text-generating neural network of any size and complexity was modified to be used on a single-domain antibody sequence dataset. This modern neural network architecture utilizes attention-weighting and skip-embedding to accelerate training and improve model quality. Furthermore, for developers, this deep learning technique can be trained to generate text for either character- or word-level.

2.4 Antibody Validation Tools

2.4.1 Perplexity (GPT model only)

Perplexity is a measurement of exponentiated average negative log-likelihood of a given input sequence. In natural language processing, tokenization is used to split texts into smaller units or "*tokens*" as assigned inputs into a language model. From the Hugging Face Artificial Intelligence Group, for a tokenized sequence $X = (x_0, x_1, \dots, x_t)$, the perplexity of X is:

$$PPL(X) = exp\left\{-\frac{1}{t}\sum_{i}^{t}\log p_\theta(x_i|x_{<i})\right\}$$

where $\log p_\theta(x_i|x_{<i})$ is the log-likelihood of the i[th] token conditioned on the preceding tokens $x_{<i}$ [11]. This measurement is used as an evaluation of the model's ability to predict uniformly among the assigned units from tokenization. In relation to ProtGPT2 model, PPL values of a generated protein can vary per sequence, and therefore sorted from the lowest to highest value as a criterion – lower values are better as the sequences resemble the nearest variant of the input protein dataset.

2.4.2 Antibody Numbering Annotation

The antibody numbering used in this GUI utilizes an online server from Andrew C R Martin's Group at University College London (https://www.bioinf.org.uk/abs/abnum/) which uses a corrected version of the Chothia scheme that was developed to be structurally correct throughout the complementarity-determining regions (CDRs) and framework regions (FRs) [14]. Commonly occurring main-chain conformations of the hypervariable regions were found to occur at sites within the hypervariable regions and in the conserved beta-sheet framework through the examination of the sequences of immunoglobulins which resulted into the Chothia Antibody Numbering annotation scheme [15]. This annotation distinguishes and outputs the sdAb's CDRs and FRs.

Chothia is a structure-based scheme, created by aligning the variable region crystal structures forming CDRs instead of a sequence-based alignment. Differences between Kabat and Chothia can be found in amino acid insertion points, for example for CDR-L1 and CDR-H1, as well as the loop lengths in CDRs [14]. Overall, Chothia numbering corresponds with the three-dimensional structures of hypervariable regions of typical length antibodies and therefore used in this study as an additional validation tool for generated sdAb sequences.

2.5 Single Domain Antibody Characteristics

Most of the biophysical properties quantified for the generated antibodies were generated from the Peptides R Package [16] which computes a set of physicochemical characteristics from the amino acid sequence. Other properties were computed through in-house calculations. These characteristics produced by the GUI are discussed in **Table 2**.

Table 2. GUI output metadata and descriptions.

Metadata	**Description**
I. Physicochemical Characteristics	
SEQ	The antibody sequence.
LENGTH	Number of amino acids in a sequence.
GRAVY_INDEX	Hydrophobicity index based on the KyteDoolittle scale [17] where higher values equate to a more hydrophobic characteristic.
MOL_WT	Single-domain antibodies are composed of a variable domain of heavy chain fragments that are around 11–15 kDa [18].
ISOELECTRIC_POINT	The pH at which the net charge of the protein is equal to 0 based on the European Bioinformatics Institute (EMBL-EBI) EMBOSS scale [19]. Proteins are positively charged at a pH below their isoelectric point and negatively charged at a pH above [20]. When the pH of the solvent is equal or higher to the isoelectric point of the protein, it tends to lose its biological function.
CYSTEINE_COUNT	Cysteine residues serve essential roles in protein structure and function by conferring stability through disulfide bond formation which maintains proper maturation and localization through protein-protein intermolecular interactions [21]. This is also important in the antibody cysteine-based conjugation in determining framework regions through numbering.
INSTABILITY_INDEX	The predicted *in vivo* stability of a protein from its primary sequence [22]. An instability index of <40 is considered as stable while >40 as unstable.
BOMAN_INDEX	The interaction index where estimates a protein's potential to bind to other proteins [23]. A high Boman index value indicates that a peptide will be multifunctional due to its ability to interact with a wide range of proteins.
CHARGE	Net charge of a protein sequence based on the Henderson-Hasselbalch equation [24] using Lehninger scale [25]. The theoretical net charge of the complementarity-determining regions (CDRs) is a strong predictor of antibody specificity or the degree to which an immune response discriminates between antigenic variants [26, 27].

II. Single Variable Domain on a Heavy Chain (VHH) Structural Characteristics

VHHs contain 4 framework regions (FRs) that form the core structure of the immunoglobulin domain and 3 complementarity-determining regions (CDRs) that are involved in antigen binding [28]. FRs are conserved regions of the antibody which allow the antigen-binding, hypervariable CDR regions to be stable [29].

H.FR1	• Framework Region 1 (N-terminus) • Possible target of efficient mutagenesis for generating a variety of affinity-matured scFv mutants [30].
H.CDR1	• Complementarity-determining Region 1 • 4 residues after first cysteine • 6-15 residues • Extended hypervariable CDR1 region (residues 27-30) is used together with CDR3 to increase surface area interacting with the antigen, and thus proposed to vary these amino acids for synthesis to increase potential antigen binding [31].
H.FR2	• Framework Region 2 • Dromedary VHHs have an extended CDR3 that is often stabilized by an additional disulfide bond with a cysteine in CDR1 or FR2 [32].
H.CDR2	• Complementarity-determining Region 2 • 10 to 20 residues after end of CDR1 • 8-15 residues • Can provide additional antigen interaction force with CDR3 [33] and an unusual extra glycine residue recently demonstrated a neutralizing spot for toxin binding [34].
H.FR3	• Framework Region 3 • Introducing a non-canonical disulfide bond into the hydrophobic core of llama VHHs between FR2 and FR3 proved to increase thermal stability at neutral pH and resistance to proteolytic degradation [35].
H.CDR3	• Complementarity-determining Region 3 • 30 to 50 residues after end of CDR2 • 3-25 residues • The most variable region and considered as the center of antigen recognition [36].
H.FR4	• Framework Region 4 (C-terminus)

3. Step-by-Step Instructions

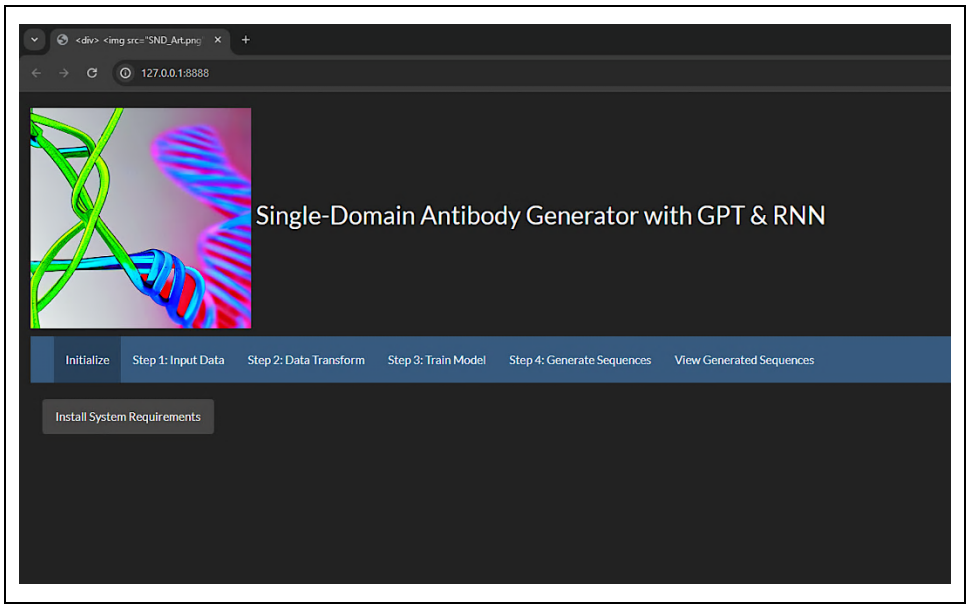

Figure 2. SDAB_GUI welcome screen.

3.1 Initialization

The GUI is set up as a ready-to-use software and comes with pre-trained GPT and RNN models used to generate novel single-domain antibodies. A 3-step process is required to initialize the GUI (**Figure 3**):

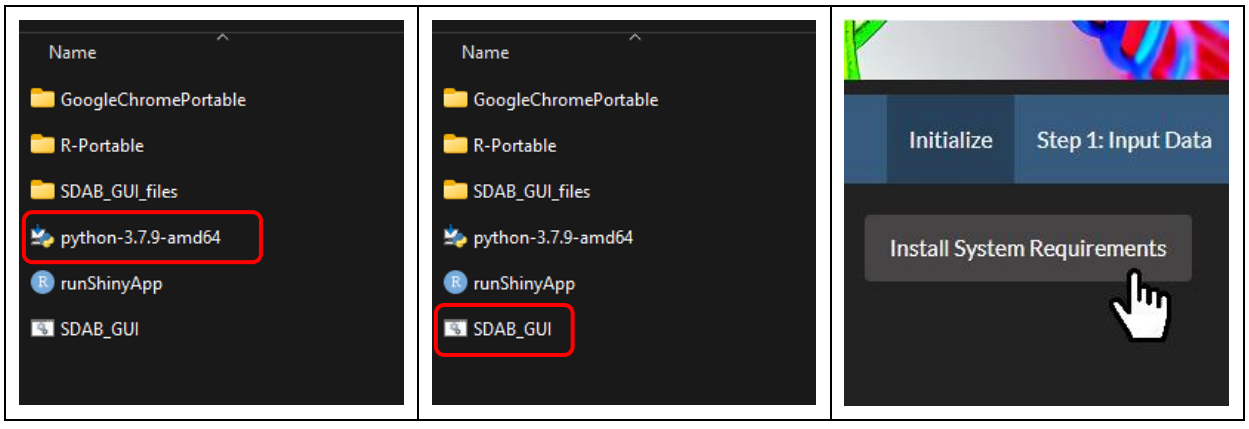

Figure 3. SDAB_GUI 3-step initialization: **1)** install Python 3.7.9; **2)** open the SDAB_GUI Windows batch file; **3)** click "Install System Requirements" when the application is running.

Upon installation, the user MUST confirm if Python 3.7.9 is added to PATH (System Environment Variables). After all the dependencies are installed, it is recommended to restart the computer.

It is possible to re-train the deep learning models if a developer decides to put additional configurations or re-parameterize either GPT, RNN, or both. Therefore, these instructions are separated into two sections: 1) for developers and 2) for front-end users.

3.2 For Developers

SDAB_GUI enables developers to input and generate novel protein sequences based on the input dataset. This series of steps (1-3) requires a basic understanding of re-parameterization into model training.

3.2.1 Step 1: Input Data

Developers can upload a CSV file containing protein sequence dataset. The default sdAb dataset used here is the "*sdab_dataset.csv*" file from the current working directory (SDAB_GUI_files folder).

- **Upload CSV File** input: CSV file requirements: the headers for the protein identifier and protein sequence shall be "*name*" and "*seq*", respectively (**Figure 4**).
- **Summary** tab: sdAb characteristics included in the summary statistics upon dataset upload.
- **Observations** tab: a table output showing the contents of the csv file. The default number of observations to view is three (3). This can be adjusted through the numeric entry labeled as "*Number of observations to view:*" above.
- **Characteristics** tab: multiple histograms showing the distribution of the sdAb sequences' physicochemical characteristics.

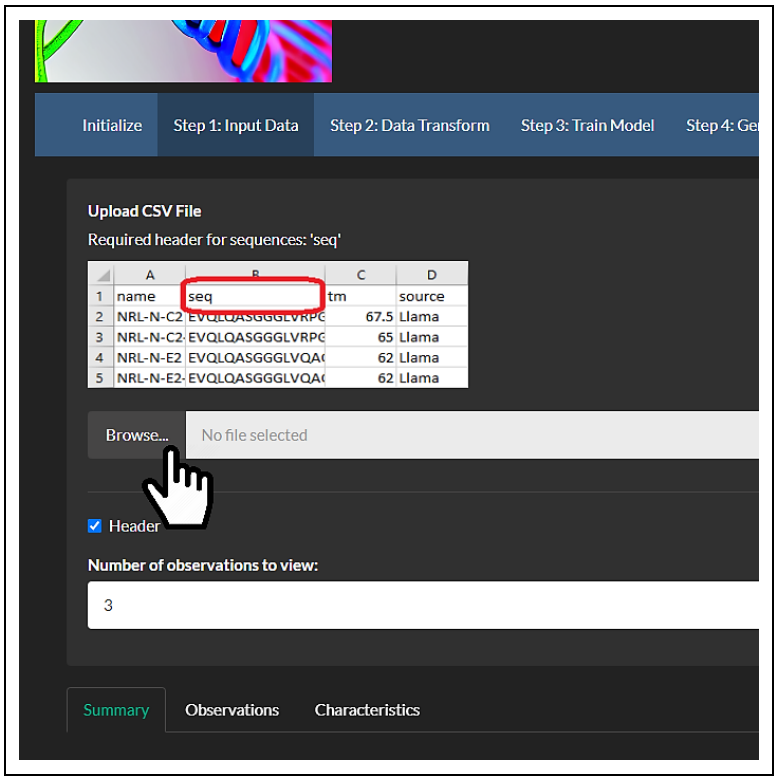

Figure 4. Step 1: Upload Data. Developers can upload an existing comma separated values (CSV) file dataset consisting of antibody sequences for model training.

3.2.2 Step 2: Transform Data

Developers transform data into training and validation test datasets for model training (**Figure 5**).

- Developers shall perform this step for model training in Step 3.
- GUI will output a warning message if no dataset was uploaded in Step 1.
- Upon transforming the dataset, GUI will output the number of sequences allocated for model training and validation datasets.
- For GPT training (90:10 split ratio): this step will output "*training.txt*" and "*test.txt*" files in the current working directory (SDAB_GUI_files folder).
- For RNN training: this step will output "*RNN_sequence_only.csv*" file.

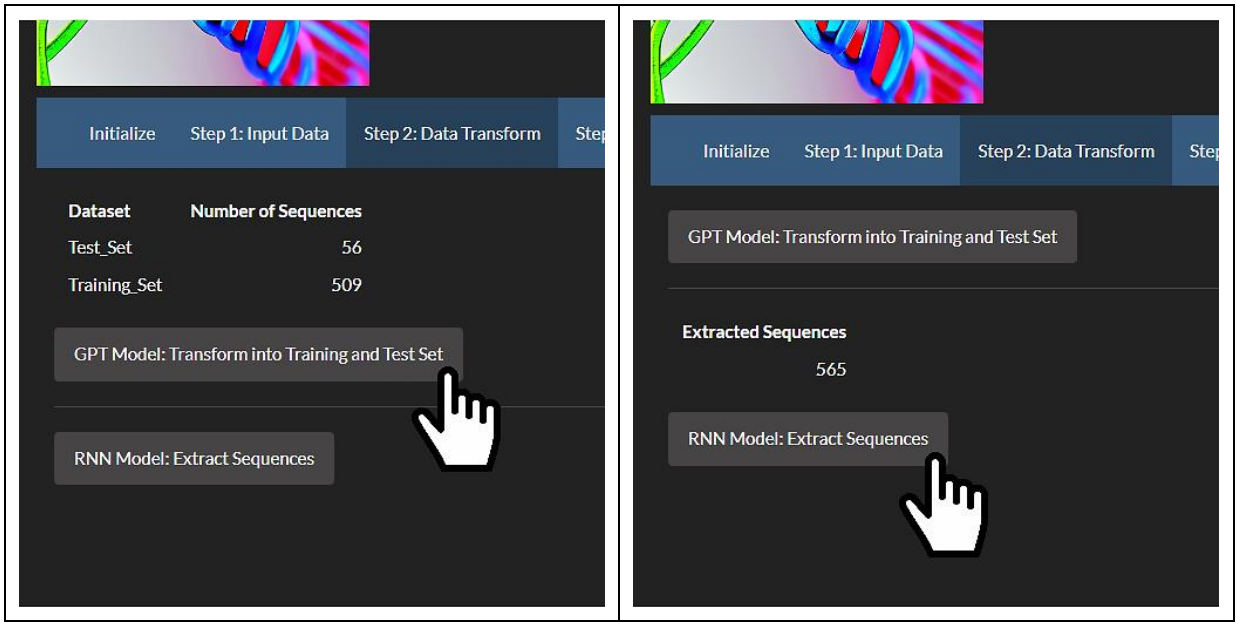

Figure 5. Step 2: Transform Data. Using the existing uploaded dataset, data transformation is required before model training. Once the transformation is complete, the GUI gives the number of sequences contained in both test and training sets.

3.2.3 Step 3: Train Model

Both GPT and RNN techniques can be trained to produce sdAb-generating models.

- This step will produce "*GPT_output_model*" and/or "*RNN_output_model*" folders in the current working directory.
- Model training can take some time to complete depending on the size of the dataset.
- This process can be skipped as the SDAB_GUI software package already has pre-trained GPT and RNN models installed.

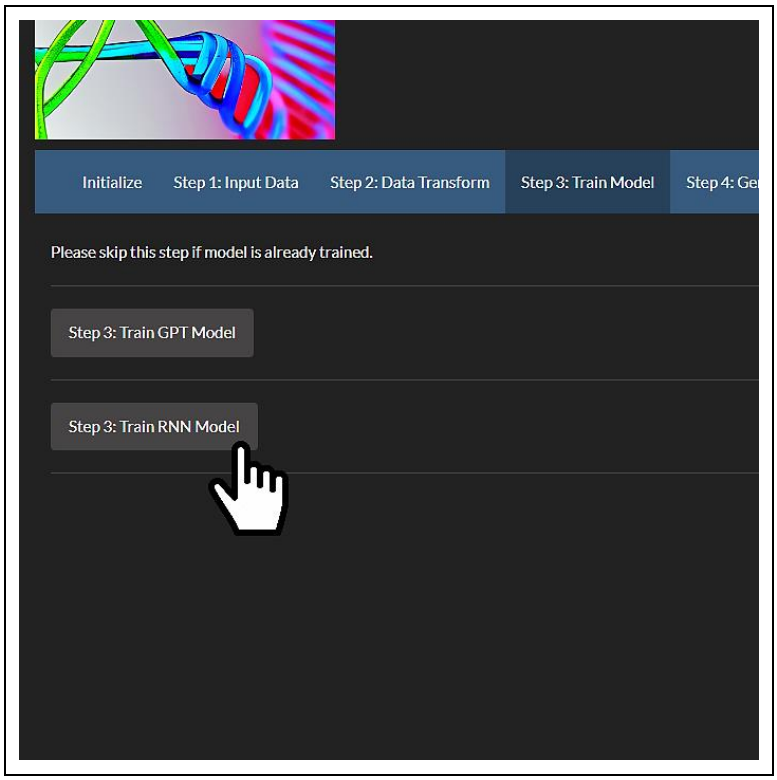

Figure 6. Step 3: GPT/RNN Model Training. A pop-up window will display the model training progress. The trained RNN or GPT model is found in the current working directory.

3.2.4 IMPORTANT INFORMATION FOR DEVELOPERS

The current working directory (SDAB_GUI_files folder) contains majority of the code that conducts the back-end process of the software. Instructions to re-parameterize the models are as follows:

GPT Model:

- Open the ***app.R*** file using a text editor software (Notepad/Notepad++) or RStudio.
- A number of parameters in line 366 can be modified (**Table 3**):
  a) *learning_rate*: default = 0.001
  b) per_device_train_batch_size: default = 4
  c) per_device_eval_batch_size: default = 4
  d) *block_size*: default = 4

Table 3. Example of GPT re-parameterization – original versus modified.

Original	*system(command = paste("python run_clm.py --model_name_or_path nferruz/ProtGPT2 --train_file training.txt --validation_file test.txt --tokenizer_name nferruz/ProtGPT2 --do_train --do_eval --output_dir*

	GPT_output_model ***--learning_rate 1e-03 --per_device_train_batch_size 4 --per_device_eval_batch_size 4 --block_size 256*** *--overwrite_output_dir"), invisible = FALSE, wait = TRUE)*
Modified	*system(command = paste("python run_clm.py --model_name_or_path nferruz/ProtGPT2 --train_file training.txt --validation_file test.txt --tokenizer_name nferruz/ProtGPT2 --do_train --do_eval --output_dir GPT_output_model* ***--learning_rate 5e-03 --per_device_train_batch_size 8 --per_device_eval_batch_size 8 --block_size 512*** *--overwrite_output_dir"), invisible = FALSE, wait = TRUE)*

RNN Model:

- Open the ***RNN_sdab_model_training.py*** file using a text editor software (Notepad/Notepad++) or any Python editor.
- A number of parameters in line 38 can be modified (**Table 4**):
  a) *rnn_size*: default = 128
  b) *dim_embeddings*: default = 100
  c) *num_epochs*: default = 5
  d) *gen_epochs*: default = 5
  e) *train_size*: default = 0.8

Table 4. Example of RNN re-parameterization – original versus modified.

Original	*textgen.train_from_file('RNN_sequence_only.csv',* *new_model=True,* ***rnn_size=128****,* ***dim_embeddings=100****,* ***num_epochs=5****,* ***gen_epochs=5****,* ***train_size=0.8****)*
Modified	... ***rnn_size=254****,* ***dim_embeddings=50****,* ***num_epochs=8****,* ***gen_epochs=8****,* ***train_size=0.5****)*

Overall, we recommend the default (original) value for these parameters as they can affect model training duration, sequence generation time, and the quality of the generated sdAbs.

3.3 For Front-End Users

SDAB_GUI enables users to generate novel antibody sequences as a ready-to-use software.

3.3.1 Step 4: Generate Sequences

Users can generate a desired number of sdAbs (**Figure 7**).

- The default number is 20 sequences.
- An output file name can be specified by the user. The default is either *GPT_generated_antibodies.csv* or *RNN_generated_antibodies.csv*.
- Due to the hardware-dependent runtimes, a loading screen is provided to see progress and a prompt will display a message upon sequence generation completion.
- Because a model can generate invalid sequences (i.e., proteins which have invalid sequence lengths, containing non-conventional amino acids, etc.), the generated sdAbs can be less than or equal to the desired number of new proteins specified by the user as they pass through several validation processes.
- The Chothia antibody annotation scheme is incorporated in the sequence generation through Antibody Numbering Tool (http://www.bioinf.org.uk/abs/abnum/) [2].
- The set of generated sequences is saved in user directory under the "*generated_sequences*" folder.
- Users can generate sdAbs through the pre-trained GPT and RNN models.

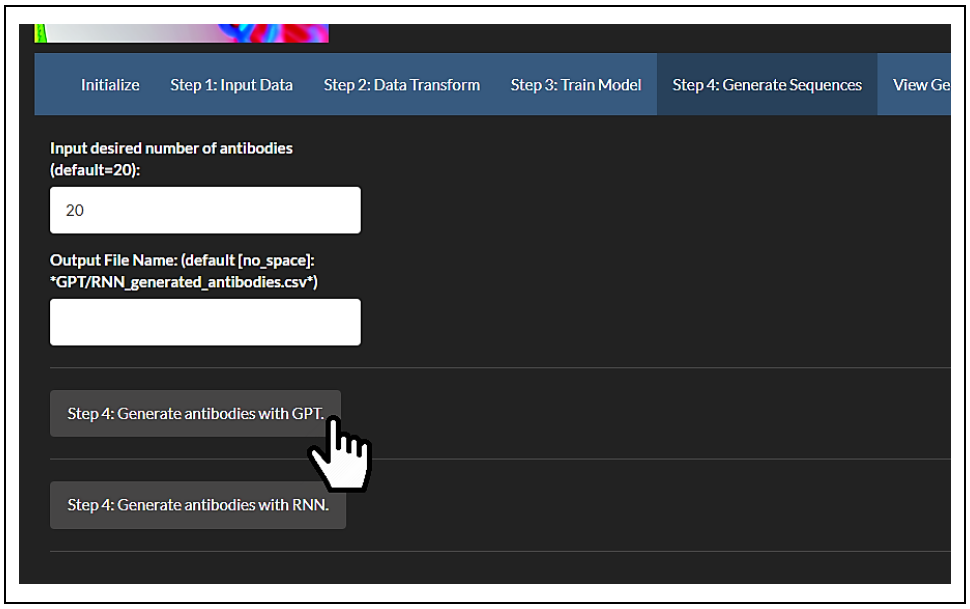

Figure 7. Step 4: Generate Proteins using the trained GPT Model. A user can specify the desired number of proteins to be generated. The default number of proteins to be generated is 20. Output file name can also be specified as selected.

3.3.2 View Generated Sequences

Users can view the generated sdAb sequences through a drop-down menu (**Figure 8**).

- **Summary** tab: sdAb characteristics included in the summary statistics upon dataset selection from the drop-down menu.
- **Observations** tab: a table output showing the contents of the csv file. The default number of observations to view is three (3). This can be adjusted through the numeric entry labeled as "*Number of observations to view:*" above.
- **Characteristics** tab: multiple histograms showing the distribution of the sdAb sequences' physicochemical characteristics.
- Upon generating another set of sdAbs while using the software, the drop-down menu can be repopulated by reloading the datasets through the "***Reload Datasets***" button (**Figure 9**). This can be also done by closing and re-opening the application through the SDAB_GUI.bat Windows batch file.
- Several other physicochemical characteristics are automatically computed based on the selected dataset from the drop-down menu (refer to previous section, ***2.5 Single Domain Antibody Characteristics***, for more information). This information can be extracted through the "***Extract Properties***" button (**Figure 10**) which saves the file under the "*generated_sequences/with_properties*" folder.

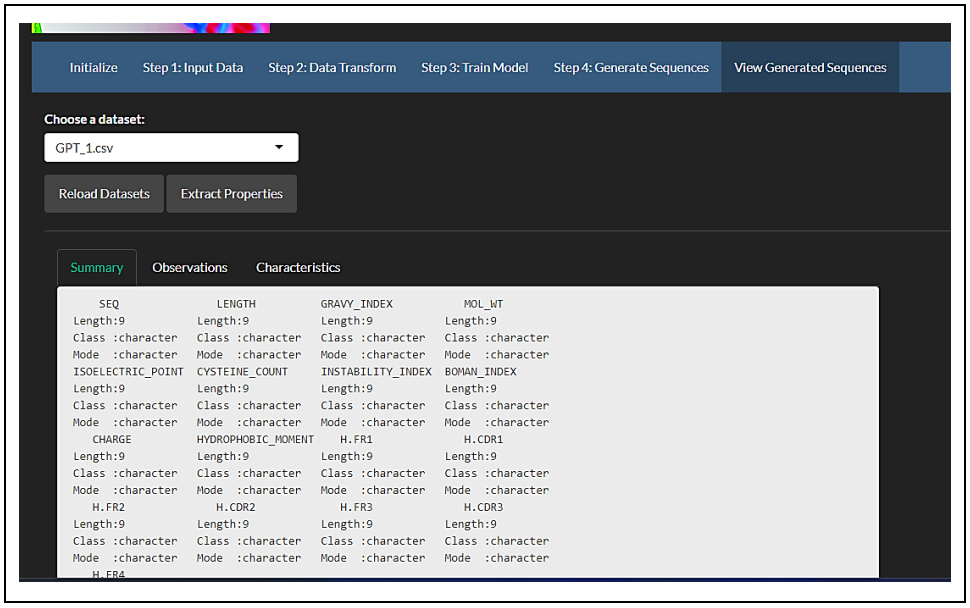

Figure 8. Generated Dataset Exploration. A drop-down menu is provided to explore generated sdAb sequences saved in "*generated_sequences*" folder under the user directory.

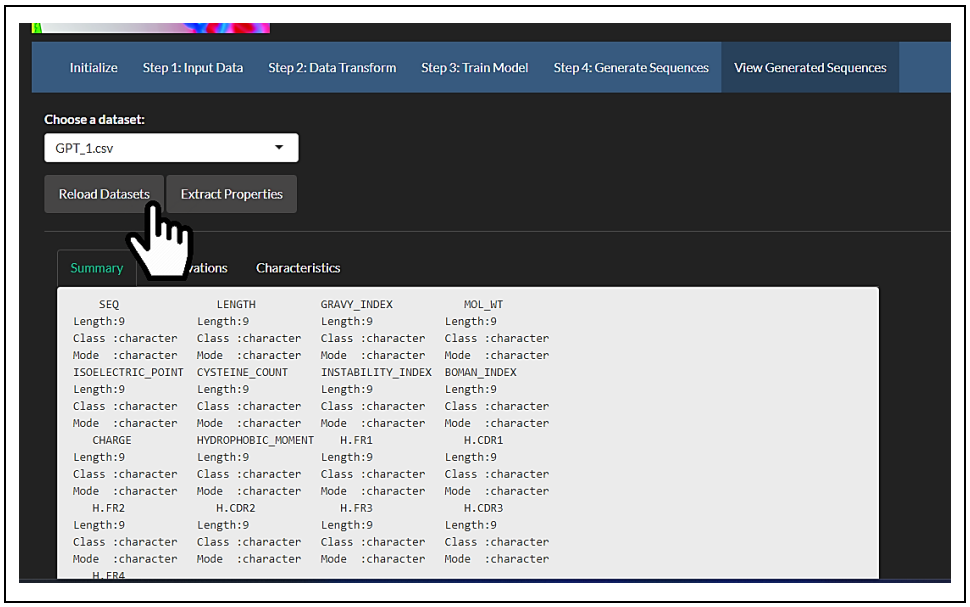

Figure 9. Reload Dataset. The "***Reload Datasets***" button will repopulate the selection of the drop-down menu in selecting the desired dataset file. This can be also done by closing and re-opening the application through the SDAB_GUI.bat Windows batch file.

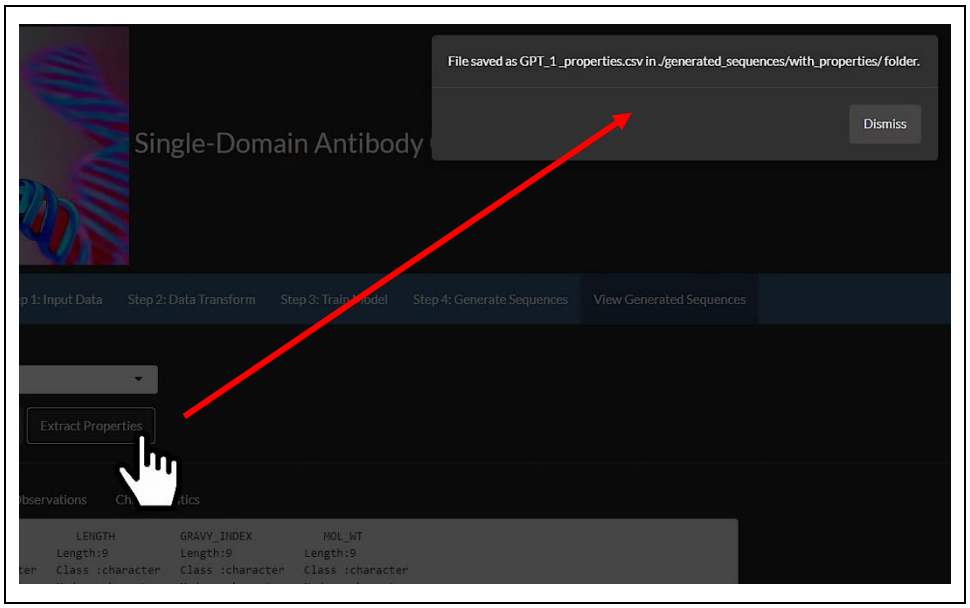

Figure 10. Extract Properties. Several physicochemical characteristics are automatically computed based on the selected dataset from the drop-down menu. This information can be extracted through the "***Extract Properties***" button which saves the file under the "*generated_sequences/with_properties*" folder.

3.4 Refresh Page

Users can perform a refresh page by clicking the refresh button on the portable Google Chrome browser or re-opening the SDAB_GUI.bat Windows batch file.

3.5 Folder Hierarchy

Parent folder and subfolders are organized as intended (**Figure 11**):

```
SDAB_GUI/
├── Google Chrome Portable
├── R-Portable
├── SDAB_GUI_files/
│   ├── generated_sequences/
│   │   ├── with_properties/
│   │   │   ├── GPT_sample1_properties.csv
│   │   │   └── RNN_sample1_properties.csv
│   │   ├── GPT_sample1.csv
│   │   └── RNN_sample1.csv
│   ├── GPT_output_model
│   ├── RNN_output_model
│   ├── app.R
│   └── ...
├── python-3.7.9-amd64.exe
├── runShinyApp.R
└── SDAB_GUI.bat
```

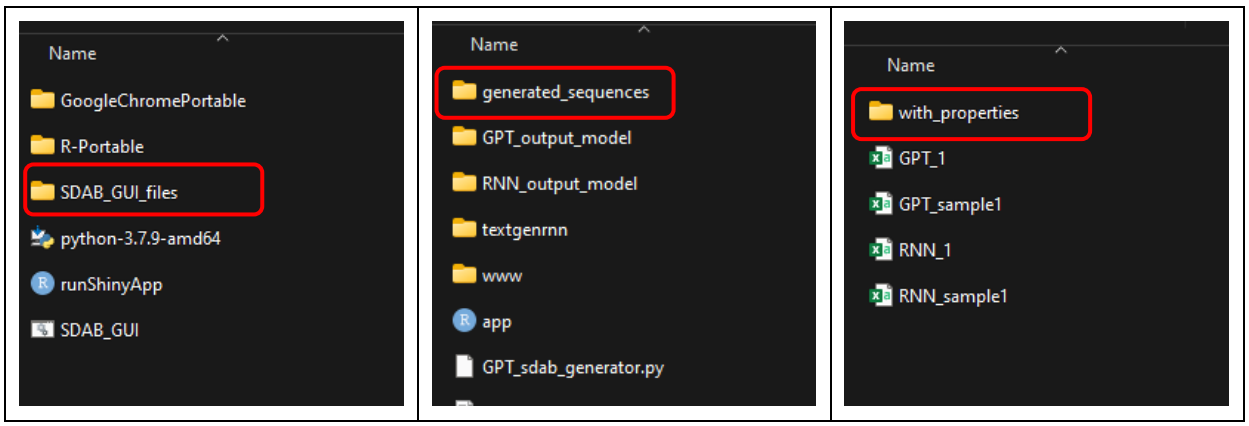

Figure 11. Folder Hierarchy. Folders and subfolders are as indicated.

3.6 Sample Generated Data

The generated sdAbs, are saved in the current working directory under the "*generated_sequences*" folder. SDAB_GUI software package comes with sample generated sdAbs for reference (**Figure 12**).

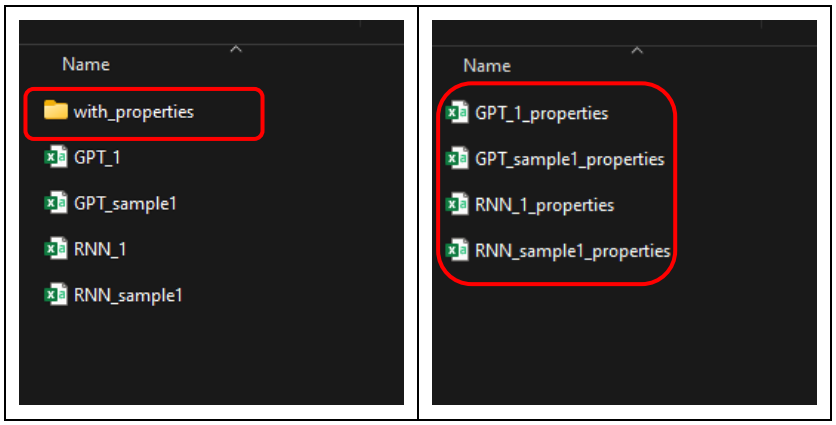

Figure 12. Sample Generated Datasets. SDAB_GUI comes with sample datasets.

4. Conclusions

Year 4 extended statement of work focused on building a fast and efficient system for creating novel sdAbs which is facilitated by the development of an easily deployable graphical user interface. The team invested on the capabilities of sdAbs with the Center's subject matter experts, curated in-house datasets, and previously reported sdAb systems that were implemented as indicated by previously reported program milestones such as the use of Generative Pre-Trained Transformers (GPT), Recurrent Neural Networks (RNN), Long Short-Term Memory – Sequence to Sequence (LSTM–Seq2Seq) and Local Interpretable Model-Agnostic Explanations (LIME) as means of generative models applied on sdAbs.

4.1 Physicochemical/Pharmacokinetic Profiles of Generated Antibodies

The generated antibodies from SDAB_GUI serve as variants from the input sdAb dataset that contains experimentally verified and naturally existing sdAb sequences from several animal sources. Although a single animal source of sequences such as camelids can be trained to provide a constraint and focused sequence profile, the output sdAbs should resemble similar or better physicochemical/pharmacokinetic properties and target antigens from their parent sequences. The basic biophysical properties such as amino acid lengths, hydrophobicity indices, cysteine counts, and instability indices of the generated sequences are specifically chosen as preliminary inspection tools for the validity of the produced sdAbs. Accompanied by the Chothia annotation scheme applied during sequence generation, the location and number of cysteines are critically important in the determination of FRs as they have several critical roles in target analytes [35]. The high stability of sdAbs indicated by the hydrophobicity and instability indices are a well-sought biophysical property which renders their potential development as a therapeutic. Parallel comparisons between the input sdAb and SDAB_GUI-generated sdAb are critical in the viability of synthetic constructs.

4.2 Future Directions

Due to the complexity of command executions in the creation of SDAB_GUI, we recommend adding an sdAb structural prediction into the generation framework such as AlphaFold [37] to further analyze the generated proteins. Moreover, several contexts from AlphaFold predictions can aid in the impact of various protein-protein or protein-ligand interactions, specifically the ability of the sdAb to bind to a specific antigen or through the plaque reduction neutralization test (e.g., PRNT50). As inclusion of AlphaFold and associated tools would have ballooned the complexity of SDAB_GUI to be run locally, especially in terms of size which would have increased by several multiples, it was left out. In future iterations, AlphaFold may be appended or a separate GUI can be introduced to be used in tandem with this sequence generator.

Acknowledgements

We acknowledge funding support through base funds of the Naval Research Laboratory (WU# 1V33) and funds from the Defense Threat Reduction Agency (HDTRA1445915, CB10869).

References

1. Ventola, C.L., *The Antibiotic Resistance Crisis.* Pharmacy and Therapeutics, 2015. **40**(4): p. 277-283.
2. Gould, I.M. and A.M. Bal, *New antibiotic agents in the pipeline and how they can help overcome microbial resistance.* Virulence, 2013. **4**(2): p. 185-191.
3. Ward, E.S., et al., *Binding activities of a repertoire of single immunoglobulin variable domains secreted from Escherichia coli.* Nature, 1989. **341**(6242): p. 544-546.
4. Hamers-Casterman, C., et al., *Naturally occurring antibodies devoid of light chains.* Nature, 1993. **363**(6428): p. 446-448.
5. Krah, S., et al., *Single-domain antibodies for biomedical applications.* Immunopharmacol Immunotoxicol, 2016. **38**(1): p. 21-8.
6. Wu, Y., S. Jiang, and T. Ying, *Single-Domain Antibodies As Therapeutics against Human Viral Diseases.* Frontiers in Immunology, 2017. **8**.
7. Samaranayake, H., et al., *Challenges in monoclonal antibody-based therapies.* Annals of Medicine, 2009. **41**(5): p. 322-331.
8. Dean, S.N., et al., *PepVAE: Variational Autoencoder Framework for Antimicrobial Peptide Generation and Activity Prediction.* Frontiers in Microbiology, 2021. **12**.
9. Stokes, J.M., et al., *A Deep Learning Approach to Antibiotic Discovery*. Cell, 2020. **180**(4): p. 688-702.e13.
10. Alvarez, J.A.E. and S.N. Dean, *Graphical User Interface for Novel Protein Generation using ProtGPT: Version 1*. 2023. p. 23.
11. Ferruz, N., S. Schmidt, and B. Höcker, *ProtGPT2 is a deep unsupervised language model for protein design.* Nature Communications, 2022. **13**(1): p. 4348.
12. Vaswani, A., et al., *Attention Is All You Need.* 2023.
13. Rumelhart, D.E., G.E. Hinton, and R.J. Williams, *Learning representations by back-propagating errors.* Nature, 1986. **323**(6088): p. 533-536.
14. Abhinandan, K.R. and A.C.R. Martin, *Analysis and improvements to Kabat and structurally correct numbering of antibody variable domains.* Molecular Immunology, 2008. **45**(14): p. 3832-3839.
15. Chothia, C. and A.M. Lesk, *Canonical structures for the hypervariable regions of immunoglobulins.* Journal of Molecular Biology, 1987. **196**(4): p. 901-917.
16. Osorio, D., P. Rondón-Villarreal, and R. Torres, *Peptides: A Package for Data Mining of Antimicrobial Peptides.* The R Journal, 2015. **7**(1): p. 4-14.
17. Kyte, J. and R.F. Doolittle, *A simple method for displaying the hydropathic character of a protein.* Journal of Molecular Biology, 1982. **157**(1): p. 105-132.
18. Holt, L.J., et al., *Domain antibodies: proteins for therapy.* Trends in Biotechnology, 2003. **21**(11): p. 484-490.
19. Madeira, F., et al., *Search and sequence analysis tools services from EMBL-EBI in 2022.* Nucleic acids research, 2022. **50**(W1): p. W276-W279.

20. Tokmakov, A.A., A. Kurotani, and K.-I. Sato, *Protein pI and Intracellular Localization.* Frontiers in Molecular Biosciences, 2021. **8**.

21. Meitzler, J.L., et al., *Conserved Cysteine Residues Provide a Protein-Protein Interaction Surface in Dual Oxidase (DUOX) Proteins.* The Journal of Biological Chemistry, 2013. **288**(10): p. 7147-7157.

22. Guruprasad, K., B.V. Reddy, and M.W. Pandit, *Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence.* Protein Engineering, 1990. **4**(2): p. 155-161.

23. Boman, H.G., *Antibacterial peptides: basic facts and emerging concepts.* Journal of Internal Medicine, 2003. **254**(3): p. 197-215.

24. Moore, D.S., *Amino acid and peptide net charges: A simple calculational procedure.* Biochemical Education, 1985. **13**(1): p. 10-11.

25. Lehninger, A.L., *Lehninger principles of biochemistry*. 6th ed. ed. 2013, New York: W.H. Freeman.

26. Rabia, L.A., et al., *Net charge of antibody complementarity-determining regions is a key predictor of specificity.* Protein Engineering, Design and Selection, 2018. **31**(11): p. 409.

27. Frank, S.A., *Specificity and Cross-Reactivity*, in *Immunology and Evolution of Infectious Disease*. 2002, Princeton University Press.

28. Harmsen, M.M. and H.J. De Haard, *Properties, production, and applications of camelid single-domain antibody fragments.* Applied Microbiology and Biotechnology, 2007. **77**(1): p. 13-22.

29. Ovchinnikov, V., et al., *Role of framework mutations and antibody flexibility in the evolution of broadly neutralizing antibodies.* Elife, 2018. **7**.

30. Kiguchi, Y., et al., *The VH framework region 1 as a target of efficient mutagenesis for generating a variety of affinity-matured scFv mutants.* Scientific Reports, 2021. **11**(1): p. 8201.

31. Nguyen, V.K., et al., *Camel heavy-chain antibodies: diverse germline V(H)H and specific mechanisms enlarge the antigen-binding repertoire*. Embo j, 2000. **19**(5): p. 921-30.

32. Muyldermans, S., et al., *Sequence and structure of VH domain from naturally occurring camel heavy chain immunoglobulins lacking light chains.* Protein Eng, 1994. **7**(9): p. 1129-35.

33. Ding, L., et al., *Structural insights into the mechanism of single domain VHH antibody binding to cortisol.* FEBS Letters, 2019. **593**(11): p. 1248-1256.

34. Rudolph, M.J., et al., *Contribution of an unusual CDR2 element of a single domain antibody in ricin toxin binding affinity and neutralizing activity.* Protein Engineering, Design and Selection, 2018. **31**(7-8): p. 277-287.

35. Bever, C.S., et al., *VHH antibodies: emerging reagents for the analysis of environmental chemicals.* Anal Bioanal Chem, 2016. **408**(22): p. 5985-6002.

36. Polonelli, L., et al., *Antibody complementarity-determining regions (CDRs) can display differential antimicrobial, antiviral and antitumor activities.* PLoS One, 2008. **3**(6): p. e2371.

37. Jumper, J., et al., *Highly accurate protein structure prediction with AlphaFold.* Nature, 2021. **596**(7873): p. 583-589.